

```
% del(X,L1,L2) :- L2 is obtained by deleting X from the list L1  
% (element,list,list) (i,i,i), nondeterm (i,i,o), nondeterm (o,i,o)
```

```
del(X,[X|Xs],Xs).
```

```
del(X,[Y|Ys],[Y|Zs]) :- del(X,Ys,Zs).
```


Student Name:

Student ID:

Section: Serial #:

| QUESTION # | 1 | 2 | 3 | 4 | 5 | 6 | TOTAL |
|---------------|----|----|----|----|----|----|-------|
| MAX POINTS | 15 | 18 | 20 | 10 | 12 | 10 | |
| POINTS EARNED | | | | | | | |

University of Bahrain

College of Information Technology
Department of Computer Science

Date: DEC 27, 2005

QUESTION ONE:

TRUE / FALSE Questions

[15 pts]

- 1) F To increment the value of K in Prolog we should use the is operator as follows: K is K+1.
- 2) F A Prolog clause always consists of a head and a body.
- 3) F Prolog proves that a given goal is true, but it cannot prove that a given goal is false.
- 4) F Having a rich set of data types and structures can significantly improve the language readability.
- 5) T The goto statement, although sometimes helps us write programs quickly, it directly reduces readability and indirectly reliability.
- 6) T The value of x in the expression $x=7-3*2$ could be 1 or 8, depending on the precedence rules specified by the grammar.
- 7) T When passing large data structures like arrays, pass-by-reference is more efficient than pass-by-value-result.
- 8) T With dynamic scoping, local variables of a program are visible to any subprograms invoked by the program.
- 9) F The visibility of dynamically scoped variables can be determined at compile time.
- 10) F The scope of a variable determines when it is allocated/deallocated.
- 11) T Dynamic binding provides programming flexibility by supporting "generic" procedures and functions that can operate on data structures with different types (for example, a "length" function on a list of elements of any type).
- 12) F Dynamic binding increases the ability to detect certain kinds of type-level errors at compile time.
- 13) T Variables of a derived type are incompatible with its parent type.
- 14) T Dynamic Type Binding increases the execution time of programs.
- 15) T Subranges of integer types are not compatible with integer types.
- 16) T Pure interpretation typically is slower than compilation.

AAA1

- 17) T A sentential form consisting of only terminals or lexemes is called a sentence.
- 18) T A subtype is compatible with its parent type.
- 19) T In a language that uses implicit type declaration, if a variable name is misspelled, it will not produce a compile-time error.
- 20) F A subprogram is active if it can be called by all other subprogram units of a program.
- 21) F Type compatibility by structure is easy to implement but highly restrictive.
- 22) F With short-circuit evaluation, $(x \neq y \ \&\& \ y \neq z)$ would not cause a run-time error, but full evaluation might cause such an error.
- 23) T The grammar that produces two different derivations for the same sentence is ambiguous.
- 24) F The binding of actual parameters to formal parameters is by position for all types of parameters.
- 25) T In C++ functions, the returned value can be of any type.
- 26) T In Ada, the parameters type of an overloaded function can be used to disambiguate calls.
- 27) T In Java, object parameters are passed by reference.
- 28) F Calling by value is more efficient (in terms of time and space) than calling by reference.

```

type indextype is 1 .. 100;
a1, a2 : integer;
b1, b2 : indextype;

```

- 29) F Variables a1 and a2 are incompatible.
- 30) T Variables a1 and b2 are incompatible.

QUESTION TWO:

Fill in blanks Questions

(21 pts)

- 1) In programming languages, a character string is defined as a primitive type (as in C# and JAVA) or as derived type (as in C++).
- ✓ 2) The main bottleneck of interpreting systems is statement decoding.
- ✓ 3) The 2 approaches for implementing the dynamic length string are: an adjacent memory cells and linked list.
- ✓ 4) The compiler must generate range checking code for every assignment to a subrange variable.
- ✓ 5) Programming languages use parentheses or brackets to indicate subscripts.
- ✓ 6) C++ allows using 3 types of arrays: static, fixed stack dynamic, and fixed heap dynamic.
- ✓ 7) The language that allows the most powerful array operations is APL.
- ✓ 8) A multi-dimensioned array in which all dimensions (rows, columns, ...) have variable number of elements is called jagged array.
- ? 9) The access function for the C++ declaration: "char a[];" is $a[i] = \text{base} + i \times \text{size}$.
- ✓ 10) In Perl, the subscript is enclosed in { } for associative arrays and [] for regular arrays.
- ✓ 11) An extra heap cell that is a pointer to the heap-dynamic variable is called tomb stone.
- ✓ 12) Pointers are used to implement managing dynamic memory and indirect addressing.
- ✓ 13) The dangling pointer problem is solved using 2 ways: tombstones and locks and keys.
- ✓ 14) The 2 automatic approaches used to reclaim garbage in the heap are: garbage collection and referencing counter.
- ✓ 15) An array in which each element is a pair of entities: a key and a value is called Associative Array.
- ✓ 16) The process that maintains a pointer counter for every heap location is called Referencing Counter.
- ✓ 17) Variables bound to storage only when they are assigned values and their attributes are bound every time they are assigned are of implicit heap dynamic type.
- ✓ 18) For the following C++ declaration: "int F[20][50];" the address of element $F[10][20] = \text{base} + ((10 \times 50) + 20) \times 4$

$$((i \times r) + j) \times 4$$

linked list, binary tree, managing dynamic memory, Tombstones, associative array, locks and keys, primitive type, referencing counters, jagged array, rectangular array, packed BCD, unpacked BCD, Access function, braces { }, parentheses (), brackets [], APL, Ada, statement decoding, CPU-memory speed gap, Perl.

QUESTION THREE:

MCQ

(20 pts)

- ✓ 1) We used EBNF to describe the _____ of programming languages.
☒ (a) syntax (b) semantics (c) all of the above (d) none
- ✓ 2) A category of languages that require the programmer to specify what is to be done only (i.e. there is no need to specify how to do things in detail).
☐ (a) imperative languages ☐ (b) procedural languages
☐ (c) scripting languages ☒ (d) logic languages
- ✓ 3) The aim of the language designer is to improve the _____ of the language.
☐ (a) readability (b) writability (c) reliability ☒ (d) all of the above
- ✓ 4) What is the meaning of a semicolon ";" in Prolog?
☒ (a) logical and (b) logical or (c) end of statement (d) none
- 5) In the following grammar, what is the associativity of the operator "#"?

$$E \rightarrow T \# E \mid T$$

$$T \rightarrow T \& \text{int} \mid \text{int} \mid (E)$$
☐ (a) left associative ☒ (b) right associative (c) All of the above (d) none
- ✓ 6) Which of the following strings cannot be generated by the grammar?

$$S \rightarrow a b S \mid c S \mid d E$$

$$E \rightarrow e E \mid e$$
☐ (a) abode ☒ (b) ababed (c) abcdce (d) ababcedde
- ✓ 7) A _____ is a category of lexemes.
☐ (a) lexicon ☒ (b) token (c) syntax graph (d) meta-language
- 8) _____ is a programming language that supports both implicit and explicit declaration of variables.
☐ (a) FORTRAN (b) Pascal (c) Java ☒ (d) none
- ✓ 9) In Prolog, the compound statement $X=5, X=X+1$.
☒ (a) always evaluates to False (i.e. gives no as an answer)
☐ (b) always evaluates to True (i.e. gives yes as an answer) and changes the value of X
☐ (c) always evaluates to True but does not change the value of X
☐ (d) gives an error.
- ✓ 10) Which kind of storage binding is needed for recursion?
☐ (a) static ☐ (b) explicit heap dynamic
☐ (c) implicit heap dynamic ☒ (d) stack-dynamic
- ✓ 11) The function that maps subscript expressions to an address in the array is called _____.
☐ (a) Unit ☒ (b) Access (c) Lambda (d) Member (e) None

AAAA

- ✓ 12) A _____ defines a collection of data objects and a set of predefined operations on those objects.
 a) data type ~~b) descriptor~~ c) variable d) array e) None
- ✓ 13) A _____ is a collection of memory cells that stores all variable attributes.
 a) data type ~~b) descriptor~~ c) variable d) array e) None
- ✓ 14) A _____ is used by compilers for type checking and building the code for allocation/deallocation operations.
 a) data type ~~b) descriptor~~ ~~c) variable~~ d) array e) None
- ✓ 15) For every assignment to a subrange variable, the _____ are checked for compatibility at compile time.
 a) accuracy b) ranges ~~c) types~~ d) balance e) None
- ✓ 16) For every assignment to a subrange variable, the _____ is done at run time.
 a) type checking ~~b) range checking~~ c) accuracy checking
 d) balance checking e) None
- ✓ 17) The code to access any array element must be generated at _____ time.
~~a) compile~~ b) run c) load d) language design e) None
- ✓ 18) The access function for the C++ declaration: "char a[10];" is $a[i] = \text{base} + \underline{\hspace{2cm}}$
 a) $i * 4$ b) $(i - 1) * 4$ c) $(i - 1) * 2$ ~~d) i~~ e) None
- ✓ 19) A procedure P1 is a dynamic parent of a procedure P2 if:
 a) P2 calls P1 b) P2 is nested in P1 ~~c) P1 calls P2~~ d) P1 is nested in P2
- ✓ 20) The meaning of language constructs: expressions, statements, and program units is called _____.
 a) Syntax ~~b) Semantics~~ c) compilation d) interpretation

QUESTION FOUR: Given the following code in an unknown block structured language.

[10 pts]

```

01 int x = 1; 2
02 int y = 2;
03 int z = 3;
04
05 int foo(int a, int b)
06 {
07     1
08     int x = 4;
09     a = a + bar(b);
10     10 2+ 8
11     2
12     return x + a + b;
13     2+10+4
14 }
15 int bar (int c)
16 {
17     x++; 2
18     c++; 4
19     return c * x;
20 }
21
22 main()
23 {
24     printf("%d", foo(y, z));
25 }

```

dynamic

static

parameters

- 1) What is the output if the language uses pass-by-value and static scoping?
✓ 15
- 2) What is the output if the language uses pass-by-value and dynamic scoping?
✓ 26
- 3) What is the output if the language uses pass-by-value-result and static scoping?
✓ 16
- 4) What is the output if the language uses pass-by-value-result and dynamic scoping?
✓ 27
- 5) What is the output if the language uses pass-by-reference and static scoping?
16
- 6) What is the output if the language uses pass-by-reference and dynamic scoping?
27

7) What is the scope of c?

function bar

OR

8) What is the scope of the x declared by the line "int x = 4;"
In inner Block of function foo() { static: line 8-9, function bar

9) What is the scope of the x declared by the line "int x = 1;"
In all programme (global variable) { except line 8-9 (static) except for 2nd (dynamic)

10) What is the scope of the y declared by the line "int y = 2;"

In all programme (global variable)

main

QUESTION FIVE:

Programming Languages Syntax:

[12 points]

- Convert the following BNF to an equivalent EBNF.

$\langle E \rangle ::= \langle E \rangle + \langle T \rangle \mid \langle T \rangle$

$\langle T \rangle ::= \langle T \rangle * \langle F \rangle \mid \langle F \rangle$

$\langle F \rangle ::= \text{id}$

$\langle E \rangle \rightarrow \langle T \rangle \{ + \langle T \rangle \}$

$\langle T \rangle \rightarrow \langle F \rangle \{ * \langle F \rangle \}$

$\langle F \rangle \rightarrow \text{id}$

- An identifier, in most programming languages, must start with a letter which can be followed by any number of letters and/or digits. Write ONE EBNF rule to describe the above definition of identifiers.

$\langle \text{ID} \rangle \rightarrow \langle \text{Letter} \rangle \{ \langle \text{Letter} \rangle \mid \langle \text{digit} \rangle \}$

- Given the grammar for expressions, answer the three questions that follow.

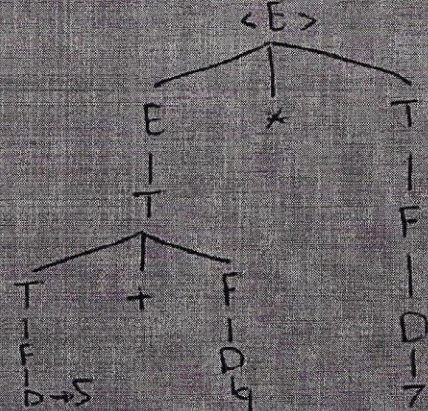
$E \rightarrow E * T \mid T$

$T \rightarrow T + F \mid F$

$F \rightarrow (E) \mid D$

$D \rightarrow 0 \mid 1 \mid 2 \mid \dots \mid 9$

- Draw a parse tree for the expression $5 + 9 * 7$



Lower is higher
Precedence than
the upper

- What does $3 * 4 + 5$ evaluate to? Give a number representing the result

(start from big one)

17 27

depend on Parse tree

- What does $3 + 4 * 5$ evaluate to? Give a number representing the result

23 35

- Write ONE EBNF rule equivalent to: $\langle \text{termlist} \rangle ::= \langle \text{term} \rangle \mid \langle \text{term} \rangle, \langle \text{termlist} \rangle$

$\langle \text{termlist} \rangle ::= \{ \langle \text{term} \rangle \}$

AAA

QUESTION SIX:

Prolog Programming:

[10 points]

What Prolog produces for each of the following queries?

✓ a) ?- [1,2|X] = [1,2,3,4,5].

X = [3,4,5]

✓ b) ?- 1+2>=1+3.

False

NO

✓ c) ?- X=5, Y=X+2.

X = 5

Y = 5+2

✓ d) ?- X is 5, Y is X+2

X = 5

Y = 7

✓ e) ?- X is 5, Y=X+2.

X = 5

Y = 5+2

what9([],0).

what9([X|Xs],S) :- what9(Xs,Ss), S is Ss + X.

f) ?- what9([1,3,5,7,9],T).

T = 25

g) ?- what9([5,6,9],U).

U = 20

• Write a Prolog predicates to count the number of elements in a list.

count([],0).

count([_|L],N) :- count(L,N1),
N is N1 + 1

Student Name:

Student id:

Sect#:A Serial #:

| QUESTION # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | TOTAL |
|------------------|---|----|----|---|----|----|----|-------|
| MAX POINTS | 8 | 18 | 15 | 8 | 10 | 10 | 12 | |
| POINTS EARNED | | | | | | | | |

University of Bahrain

Department of Computer Science

College of Information Technology

ITCS332: Concepts of Programming Languages FINAL TEST

Date: JUNE 16, 2007

QUESTION ONE: True/False Questions: (Use F for true and T for false)

[8 pts]

- 1) T Ada allows the programmer to choose normal or short-circuit evaluation of boolean expressions.
- 2) T A variable of a derived type is incompatible with its parent type.
- 3) T A subtype variable is compatible with its parent type.
- 4) T A sentential form consisting of only terminals or lexemes is called a sentence.
- 5) F Java programs are less reliable than semantically equivalent C++ programs.
- 6) F ^{static(T)} Dynamic scope is based on the spatial textual layout of program units.
- 7) F Having a rich set of data types and structures can significantly reduce compilation time.
- 8) F Row-major ordering of an array occupies less space than column-major ordering..
- 9) T The grammar that produces two different derivations for the same sentence is ambiguous.
- 10) T In most languages, all numeric types in expressions are coerced using widening conversions.
- 11) T The value of x in the expression: $x = 7 - 3 * 2$ could be 1 or 8, depending on the precedence rules.
- 12) T With dynamic scoping, local variables of a main program are visible to any active subprogram.
- 13) F Dynamic Type Binding reduces the execution time of programs.
- 14) T Subranges of integer types are not compatible with integer types
- 15) F Dynamic binding increases the compiler ability to detect certain kinds of type-level errors.
- 16) F To increment the value of K in Prolog we should use the "is" operator as follows: K is K+1.

QUESTION TWO:

Fill in blanks Questions (See some answers below)

[18 pts]

- 1) Functional side effects occur when a function changes a two-way parameter or non local variable
- 2) "int *U=NULL; cout << *U;" suffers from Dereferencing uninitialized problem.
- 3) "int const k =99; char h[k];" defines a limited dynamic length string.
- 4) In PHP, the "2" == 2 produces False, while the "2" === 2 produces True.
- 5) Variables bound to storage only when they are assigned values and their attributes are bound every time they are assigned are of implicit heap dynamic type.
- 6) In denotational semantics, mathematical objects are used to represent the meaning of language constructs.
- 7) The value of Synthesized attribute on a parse tree depends only on the values of the attributes on that node's children nodes.
- 8) In mixed-mode assignments, the coercion takes place only after the RHS expression has been evaluated or another alternative is to coerce all operand in RHS to the type of the target before evaluation.
- 9) To solve ambiguities in arithmetic expressions, language designers include static semantic EBNF rules in the language syntax.
- 10) A multi-dimensioned array in which all dimensions (rows, columns, ...) have variable number of elements is called Jagged array
- 11) An extra cell that is a pointer to the actual heap-dynamic variable is called tombstone
- 12) The dangling pointer problem is solved using 2 ways: tombstones and Locks and keys
- 13) The compiler must generate Range checking code for every assignment to a subrange variable.
- 14) The 2 automatic approaches used to reclaim garbage in the heap are: garbage collection and reference counter
- 15) A Short circuit Evaluation is one in which the result of an expression is determined without evaluating all of the operands and/or operators.
- 16) For the following C++ declaration: "int F[200][150];" if the base address of array F is 2400, then the address of element F[80][32] = 2400 + 4 * [(80 * 150) + 32]
- 17) The access function for the C++ array declaration: "double t[100];" is t[75] = base + 8 * 75
- 18) "int a=22,b=4; float c; c=a/b;" if operand coercion is done before assignment coercion the value of c will be 5.5

linked list, binary tree, CPU-memory speed gap, indirect addressing, tombstones, associative array, braces {}, Perl, locks and keys, primitive type, reference counters, jagged array, rectangular array, packed BCD, unpacked BCD, Access function, parentheses (), brackets [], APL, Ada, dangling pointers, statement decoding, denotational, operational, axiomatic, inherited, synthesized.

QUESTION THREE:

MCQ

[15 pts]

- 1) An example non-associative programming language is: 2)
 (a) C++ (b) APL (c) Java (d) Ada (e) none
- 2) What is the meaning of a comma “,” in Prolog?
 (a) logical and (b) logical or (c) end of statement (d) logical not (e) none
- 3) A _____ defines a collection of data objects and a set of predefined operations on those objects.
 a) array (b) descriptor (c) variable (d) data type (e) None
- 4) The function that maps subscript expressions in the array reference to an address is called _____.
 a) Unit (b) Negation (c) Access (d) Member (e) None
- 5) A procedure P1 is a static parent of a procedure P2 if:
 a) P2 calls P1 (b) P2 is nested in P1 (c) P1 calls P2 (d) P1 is nested in P2
- 6) Which of the following strings cannot be generated by the grammar?
 $\langle S \rangle \rightarrow a b \langle S \rangle \mid c \langle S \rangle \mid d \langle E \rangle$
 $\langle E \rangle \rightarrow e \langle E \rangle \mid e$
 a) abcde (b) ababccdee (c) abcdee (d) ababacd (e) none
- 7) A procedure P1 is a dynamic parent of a procedure P2 if:
 (a) P2 calls P1 (b) P2 is nested in P1 (c) P1 calls P2 (d) P1 is nested in P2
- 8) _____ is a programming language that supports implicit declaration of variables.
 (a) FORTRAN (b) Pascal (c) Java (d) C++ (e) none
- 9) Which kind of storage binding is needed to support recursion?
 a) static (b) explicit heap dynamic (c) implicit heap dynamic (d) stack-dynamic
- 10) One of the following is NOT harmful for language readability.
 a) Operator Overloading (b) Feature multiplicity (c) Presence of Boolean type (d) Large number of basic constructs (e) None
- 11) A category of languages in which computations are mainly specified with variables and assignments is
 (a) Markup (b) Functional (c) logical (d) imperative (e) Object-oriented
- 12) A _____ is a collection of memory cells that stores all variable attributes.
 a) data type (b) descriptor (c) Scope (d) Lifetime (e) None
- 13) The code to access any array element must be generated at _____ time.
 (a) compile (b) run (c) load (d) language design (e) None
- 14) A _____ is used by compilers for type checking and building the code for allocation /deallocation operations
 a) data type (b) descriptor (c) variable (d) array (e) None
- 15) For every assignment to a subrange variable, the _____ are checked for compatibility at run time.
 a) accuracy (b) ranges (c) types (d) balances (e) None

| Question # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Answer # | | | | | | | | | | | | | | | |

[8 pts]

```

01 int x = 1;
02 int y = 2;
03 int z = 3;
04
05 int foo(int a, int b)
06 {
07 {
08     int x2 = 4;
09     a = a2 + bar(b);
10 }
11     4 + 6 + 4
12     return x + a + b;
13 }
14     4 + 10 + 3 = 17
15 int bar (int c)
16 {
17     x++;
18     c++;
19     return c * x;
20 }
21
22 main()
23 {
24     printf("%d", foo(y, z));
25 }

```

- 15
- 2) What is the output if the language uses pass-by-value and dynamic scoping?

- 2) What is the output if the language uses pass-by-value and dynamic scoping?

- 3) What is the output if the language uses pass-by-reference and static scoping?

- 4) What is the output if the language uses pass-by-reference and dynamic scoping?

- 5) What is the scope of c?

- 6) What is the scope of the x declared by the line "int x = 4;?"

- 7) What is the scope of the x declared by the line "int x = 1;"?

- 8) What is the scope of the y declared by the line "int y = 2; "?

QUESTION FIVE:

Programming Languages Syntax:

[10 points]

- Write ONE EBNF rule equivalent to the following BNF rule: $\langle \text{expList} \rangle ::= \langle \text{exp} \rangle \mid \langle \text{exp} \rangle ; \langle \text{expList} \rangle$

* $\langle \text{expList} \rangle ::= \{ \langle \text{exp} \rangle \}$ $\langle \text{expList} \rangle ::= \langle \text{exp} \rangle \{ \langle \text{exp} \rangle \}$

- Convert the following BNF rules into an equivalent EBNF.

$\langle E \rangle ::= \langle E \rangle * \langle T \rangle \mid \langle E \rangle \% \langle T \rangle \mid \langle T \rangle$
 $\langle T \rangle ::= \langle T \rangle - \langle F \rangle \mid \langle T \rangle + \langle F \rangle \mid \langle F \rangle$
 $\langle F \rangle ::= \langle \text{id} \rangle \mid \langle \text{int} \rangle$

$\langle \text{expList} \rangle ::= \langle \text{exp} \rangle \{ \langle \text{exp} \rangle \}$

$\langle E \rangle ::= \{ \langle E \rangle * \langle T \rangle \} \langle T \rangle$
 $\langle T \rangle ::= \{ \langle T \rangle - \langle F \rangle \} \langle F \rangle$
 $\langle F \rangle ::= \langle \text{id} \rangle \mid \langle \text{int} \rangle$

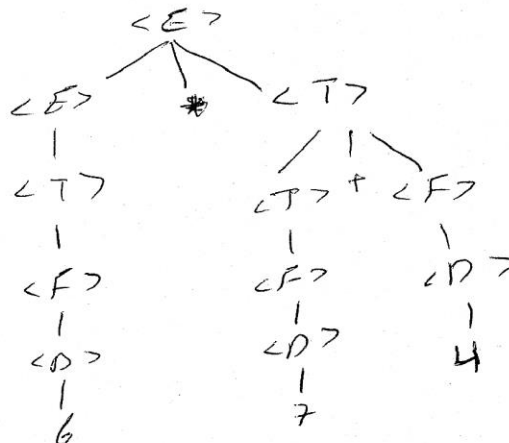
- A signed float value $\langle \text{float} \rangle$ is a sequence of a sign (+ or -) followed by a sequence of digits followed by dot followed by a sequence of digits. Examples of accepted values: -7.6, 307.80, -0.2, +20,34567, ... Write ONE EBNF rule to describe the above definition of $\langle \text{float} \rangle$ values.

$\langle \text{float} \rangle ::= [(+ | -)] \langle \text{digit} \rangle \{ \langle \text{digit} \rangle \} [. \langle \text{digit} \rangle \{ \langle \text{digit} \rangle \}]$

- Given the grammar for expressions, answer the three questions that follow:

$\langle E \rangle ::= \langle E \rangle * \langle T \rangle \mid \langle T \rangle$
 $\langle T \rangle ::= \langle T \rangle + \langle F \rangle \mid \langle F \rangle$
 $\langle F \rangle ::= (\langle E \rangle) \mid \langle D \rangle$
 $\langle D \rangle ::= 0 \mid 1 \mid 2 \mid \dots \mid 9$

- Draw ALL parse trees that can be constructed for the expression $6 * 7 + 4$?



- What does $4 * 6 + 5$ evaluate to? Give a number representing the result.

44

- What does $4 + 6 * 5$ evaluate to? Give a number representing the result.

50

QUESTION SIX:

Prolog Programming

[6+4=10 points]

What Prolog produces for each of the following queries?

a) ?- [1, 2 | X] = [1, 2, 3, 4, 5].

$X = [3, 4, 5]$

b) ?- 1+2>=1+3.

~~Yes~~ No

c) ?- X is 15, Y=X+20.

$X = 15$

$Y = 15 + 20$

d) ?- X is 50, Y is X*2

$X = 50$

$Y = 100$

what9([], 0).

what9([A|B], F) :- what9(B, C), F is C + A.

f) ?- what9([3.5, 4.75, 2.75], T).

$T = 11$

g) ?- what9([5, 6, 9], U).

$U = 20$

- Write Prolog predicates named "del" to delete a given item X from a given list named L1 to produce a new list named L2.

$del(X, L1)$

QUESTION SEVEN:

Variable Types

[12 points]

The following program illustrates different types of variables used in C++ (static, ...). Carefully study the program and answer the questions below.

```

1) #include <iostream>
2) using namespace std;
3) void tester()
4) {
5)     static int stat = 3;   int aut = 0;
6)     stat += stat;          aut+=5;
7)     cout << stat << "\t" << aut << endl; }
8) int * getloc()
9) {
10) int * temp = new int;   return temp; }
11) void loseloc(int * ptr)
12) {
13)     cout << endl;   delete ptr; }
14) void main()
15) {
16)     for (int i = 1; i <= 3; i++)   tester();   cout << endl;
17)     int * ptr = getloc();
18)     *ptr = 100;
19)     cout << *ptr << endl << endl;
20)     loseloc(ptr);
21)     int * p2 = new int;
22)     cout << *p2 << endl << endl;
23)     *ptr = 250;
24)     cout << *ptr << "\t" << *p2 << endl;
25) }
    
```

- a) What will be printed after executing the entire loop in line 16? 6 5
12 5
24 5
- b) The lifetime of a variable `stat` begins when function begins and ends when function ends.
- c) The type of a pointer `temp` defined in line 10 is stack dynamic.
- d) The type of a variable pointed to by `temp` defined in line 10 is Explicit heap dynamic.
- e) The type of a variable `aut` defined in line 5 is stack dynamic.
- f) The lifetime of a variable pointed to by `temp` begins when temp is created and ends when temp is deleted.
- g) The lifetime of a variable `aut` begins when function begins called and ends when function begins terminated.